## REMARKS

Applicant is in receipt of the Office Action mailed July 24, 2007. Claims 2-30 and 32-40 were pending in the application and were rejected. Claims 5-10, 13-18, 22-24, 30, 32, 33, and 35-40 have been amended. Claims 19-21, 25-29, and 34 have been canceled. Claims 2-18, 22-24, 30, 32-33, and 35-40 remain pending in the application. Reconsideration of the case is earnestly requested in light of the following remarks.

Section 112 Rejections

Claims 6-10 and 21-22 were rejected under 35 U.S.C. 112, second paragraph, as having insufficient antecedent basis for certain limitations recited in these claims. Applicant respectfully submits that the claim amendments overcome these rejections.

Claims 16-24 were rejected under 35 U.S.C. 112, first paragraph, as containing subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Applicant respectfully submits that the claim amendments overcome these rejections.

Section 102 Rejections

Claims 30 and 32-40 were rejected under 35 U.S.C. 102(a) as being anticipated by Keller et al., "Dynamic Dependencies in Application Service Management." Applicant respectfully traverses these rejections.

Claim 30 recites in pertinent part:

> automatically discovering a first component and a second component in the IT system, wherein automatically discovering the first component comprises automatically discovering that one or more elements of the first component are present in the IT system, wherein automatically discovering the second component comprises automatically discovering that one or more elements of the second component are present in the IT system;

Keller relates generally to dependency analysis in an application service management

system, e.g., analyzing dependencies between components in an application service management system. However, Keller does not teach automatically discovering the components for which the dependencies are analyzed. In particular, Keller does not teach automatically discovering a first component (or a second component) in an IT system, wherein automatically discovering the first component (or the second component) comprises automatically discovering that one or more elements of the first component (or the second component) are present in the IT system. Applicant thus respectfully submits that claim 30 is patentably distinct over Keller for at least this reason.

Claim 30 also recites the further limitations of:

> monitoring the usage of resources by the discovered first and second components in the IT system by receiving real-time messages;
>
> in response to receiving a first real-time message indicating that the first component uses a particular resource, sending a first resource usage message to an accumulator, wherein the first resource usage message indicates that the first component uses the particular resource;
>
> in response to receiving a second real-time message indicating that the second component uses the particular resource, sending a second resource usage message to the accumulator, wherein the second resource usage message indicates that the second component uses the particular resource;
>
> the accumulator indicating that a first dependency between the first component and the second component exists in response to determining that the first and second resource usage messages indicate that the first component and the second component both use the particular resource;

Applicant respectfully submits that Keller does not teach these limitations. Keller emphasizes repeatedly throughout the disclosure that dependencies are statically analyzed by analyzing information stored in repositories. See, for example, the following passages of Keller's disclosure:

> The approach for identifying and computing dependencies presented in this paper is pragmatic and based on a **static dependency analysis** that yields information on entities within a system (Intrasystem) and between peer entities of a service (Intersystem). [Section 6: Conclusion and Outlook, p. 7, right column, second paragraph]

> Considering the fact that a majority of application services run on UNIX and Windows NT-based systems, it is worth analyzing the degree to which information regarding applications and services is already contained in the operating systems. The underlying idea is as follows: if it is possible to obtain a reasonable amount of

information from these sources, the need for application-specific instrumentation can be greatly reduced. Our approach recognizes the fact that system administrators successfully deploy applications and services without having access to detailed, application-specific management instrumentation.

Windows NT/95/98 systems and UNIX implementations such as IBM AIX and Linux have built-in repositories that keep track of the installed software packages, filesets and their versions. AIX Object Data Manager (ODM), Windows Registry, and Linux Red Hat Package Manager (RPM) are examples for these system-wide configuration repositories. In this paper, we will concentrate on ODM. However, we have verified the applicability of our approach to the other repositories as well. [Section 4: Dependency Analysis, p. 5, bottom of left column to top of right column]

Our analysis has shown that system repositories such as ODM represent a rich source of application service management information, not only regarding the configuration of installed applications but also for determining dependency relationships between applications and services. Note that this large amount of information can be obtained *without any* specific instrumentation of the components. The only requirement is that the application components be described using a service description template that has been developed by us and whose content can be provided to a large degree by today's system information repositories. [Section 4: Dependency Analysis, p. 6, left column]

We assume that through **offline analysis – based on the methodology and approach presented in section 4** – a database of static dependencies is constructed. This collected data describes, for each end-to-end application service, the dependencies it has on lower level application and network layer services and components. [Section 5: Dependency Architecture, p. 6, right column]

From the passages quoted above, it is very clear that Keller's method for determining dependency relationships involves the static, offline analysis of system repositories. In contrast, the dependency between the first component and the second component in claim 30 is indicated by the accumulator in response to determining that the first and second resource usage messages indicate that the first component and the second component both use the particular resource, wherein the first resource usage message is sent to the accumulator in response to receiving a first real-time message indicating that the first component uses the particular resource, and wherein the second resource usage message is sent to the accumulator in response to receiving a second real-time message indicating that the second component uses the particular resource. Keller simply does not teach these limitations in combination with the other limitations recited in

claim 30.

Applicant thus respectfully submits that claim 30 is patentably distinct over Keller for at least the reasons set forth above. Inasmuch as claim 40 recites similar limitations as claim 30, Applicant respectfully submits that claim 40 is also patentably distinct over Keller.

Section 103 Rejections

Claims 2-12 and 14-15 were rejected under 35 U.S.C. 103(a) as being unpatentable over Kar et al., "An Architecture for Managing Application Services over Global Networks" in view of Kathrow et al., U.S. Patent No. 6,393,438 B1. Applicant respectfully traverses these rejections.

Claim 5 recites in pertinent part:

> creating a plurality of component fingerprints, wherein the plurality of component fingerprints includes a fingerprint for a first component, wherein creating the fingerprint for the first component comprises identifying a plurality of attributes of the first component and selecting one or more, but not all, of the plurality of attributes as the fingerprint for the first component;
>
> automatically discovering the existence of a plurality of components in an information technology (IT) system using the plurality of component fingerprints, wherein the discovered components include the first component, wherein automatically discovering the existence of the first component comprises:
>
> receiving a plurality of event messages indicating a plurality of real-time events that occur in the IT system, wherein each event message matches a respective attribute of the fingerprint for the first component; and
>
> determining that event messages matching every attribute of the fingerprint for the first component have been received;

Regarding the use of component fingerprints as recited in claim 5, the Examiner cites Kathrow's teaching regarding the use of file fingerprints or signatures. Kathrow teaches determining a hash of a file and a size of the file, which are two 4-byte values that are together referred to as a "fingerprint" for the file. However, Kathrow does not teach the limitations recited in claim 5 of creating a fingerprint for a first component, wherein creating the fingerprint for the first component comprises identifying a plurality of attributes of the first component and selecting one or more, but not all, of the plurality of attributes as the fingerprint for the first

component.

Furthermore, Kathrow teaches the use of a file fingerprint (also referred to as a signature – See Col. 10, lines 16-18 and Abstract) to identify the existence of differences between two files, but Kathrow does not teach the use of a fingerprint for a first component to automatically discover the existence of the first component in an IT system. More particularly, Kathrow does not teach or even remotely suggest the recited limitations of:

> wherein automatically discovering the existence of the first component comprises:
>
> receiving a plurality of event messages indicating a plurality of real-time events that occur in the IT system, wherein each event message matches a respective attribute of the fingerprint for the first component; and
>
> determining that event messages matching every attribute of the fingerprint for the first component have been received;

Applicant also notes that the problem of identifying the existence of differences between two files is clearly not at all the same as the problem of automatically discovering the existence of a first component in an IT system and respectfully submits that Kathrow is not analogous art with respect to claim 5.

Thus, Applicant respectfully submits that claim 5 is patentably distinct over the cited references for at least the reasons set forth above. Inasmuch as claims 14 and 15 recite similar limitations as claim 5, Applicant respectfully submits that claims 14 and 15 are also patentably distinct over the cited art.

Applicant notes that Kathrow was also used as a reference in rejecting the other independent claims 13, 16, and 23. Inasmuch as these claims recite similar limitations as those discussed above regarding the use of component fingerprints to automatically discover components in an IT system, Applicant submits that similar arguments as set forth above with reference to Kathrow also apply to these independent claims and that it is abundantly clear that Kathrow, taken either singly or in combination with the other cited references, does not teach the limitations recited in these claims.

Applicant thus respectfully submits that all of the independent claims are patentably distinct over the cited art. Since the independent claims have been shown to be patentably distinct over the cited art, Applicant respectfully submits that the dependent claims are also patentably distinct for at least this reason. Applicant also submits that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the independent claims have been shown to be patentably distinct, a further discussion of the dependent claims is not necessary at this time.

**CONCLUSION:**

Applicants submit the application is in condition for allowance, and an early notice to that effect is requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above-referenced application from becoming abandoned, Applicant hereby petitions for such extension.

The Commissioner is authorized to charge any fees that may be required, or credit any overpayment, to Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C. Deposit Account No. 501505/6002-07000/DMM.

Respectfully submitted,

Date: October 24, 2007

By: /Dean M. Munyon/
Dean M. Munyon
Reg. No. 42,914

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P. O. Box 398
Austin, Texas 78767
(512) 853-8847